

AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions, and listings, of claims in the captioned patent application:

Listing of Claims:

1. (Currently Amended) A method of monitoring response time of a method or a function associated with a Java software component, comprising the steps of:

inserting an instrumentation code in a bytecode representation of said method or function to generate a call to an Application Response Measurement (ARM) agent to cause the agent to effect generation of a start time marker upon start of execution of said method or function and a stop time marker upon completion of execution of said method or function, wherein the ARM agent is one of a plurality of agents of an ARM protocol; and

utilizing said start and stop time markers to determine a response time of said method or function.

2. (Canceled)

3. (Original) The method of claim 2, further comprising:

registering said method or function with said ARM agent prior to generation of said start and stop time markers.

4. (Original) The method of claim 2, wherein said instrumentation code causes generation of said start and stop time markers without modifying instructions associated with execution of said method or function.

5. (Original) The method of claim 2, wherein said ARM agent generates a record corresponding to said method or function for storing the response time associated with said method or function.

6. (Original) The method of claim 5, wherein said record includes a field for identifying a parent, if any, of said method or function in a hierarchical parent-child transaction chain.

7. (Original) The method of claim 6, wherein said record includes another field for identifying a top level transaction in said parent-child transaction chain.

8. (Currently Amended) The method of claim 1, wherein said software component can be any of a server page JSP, ~~a servlet~~ a servlet of a server side component EJB, a JDBC driver, a naming and directory interface (NDI) ~~JNDI~~ or ~~an RMI~~ remote method invocation (RMI) component.

9. (Currently Amended) The method of claim 8, wherein said ~~method of~~ method or function of the software component comprises any of a ~~jspService~~ method of a server page JSP, a doFilter, a doGet, a doPost or a service method of a servlet ~~servlet~~, a getConnection, executeQuery, or selected methods of JDBC driver, or remote, local or home interface methods of a server side component. ~~an EJB.~~

10. (Currently Amended) The method of claim 1, wherein the step of inserting the instrumentation code comprises incorporating instrumentation hooks into said bytecode representation prior to or during loading and initialization of a class containing said method or function by a virtual machine ~~Java Virtual Machine (JVM)~~.

11. (Original) The method of claim 1, further comprising:
storing said response time in a database.

12. (Original) The method of claim 1, further comprising:
displaying said response time to a user.

13. (Currently Amended) A system comprising a processor configured to monitor a ~~for~~ monitoring response time of a method or function associated with a ~~Java~~ software component, said processor configured to implement at least: comprising:

an instrumentation engine for inserting instrumentation code in a ~~byte code~~ bytecode representation of said method or function, said instrumentation code effecting generation of a start time marker and a stop time marker upon resumption and completion, respectively, of said method or function,

an interface module being invoked by said instrumentation code upon start and completion of said method or function, ~~and~~

an application response measurement (ARM) agent in communication with said interface module,

wherein said interface module, upon invocation by said instrumentation code, calls said ARM agent to cause generation of said start and stop time markers by said ARM agent, and wherein the ARM agent is one of a plurality of agents of an ARM protocol; and

an analysis and presentation module in communication with said ARM agent for presenting said response time to a user and/or storing said response time in a database.

14. (Currently Amended) The system of claim 13, wherein said instrumentation engine inserts said instrumentation code prior to loading of a class containing said method or function by a virtual machine ~~Java Virtual Machine (JVM)~~.

15. (Original) The system of claim 13, wherein said instrumentation engine inserts said instrumentation code in said bytecode representation without modifying instructions associated with execution of said method or function.

16. (Canceled)

17. (Currently Amended) A method of measuring a response time of a transaction initiated by a web server in response to an HTTP request, comprising:

utilizing receiving, by a monitoring agent, a first call back provided by the web server indicating that the transaction has commenced;

invoking, by the monitoring agent in response to receiving the first call back, a start method of an Application Response Measurement (ARM) agent to save a start time marker upon start of said transaction, wherein the ARM agent is one of a plurality of agents of an ARM protocol;

utilizing receiving, by the monitoring agent, another a second call back from the web server indicating that the transaction is completed;

invoking, by the monitoring agent in response to receiving the second call back, a stop method of the ARM agent to save a stop time marker upon completion of said transaction[[,]]; and

utilizing said start and stop time markers ~~to measure a~~ to measure the response time associated with said transaction.

18. (Canceled)

19. (Currently Amended) The method of claim 18, further comprising:
registering said monitoring agent with said web server to receive said call backs.

20. (Currently Amended) The method of claim 19, further comprising:
transmitting said response time to a presentation module for presentation to a user.

21. (Currently Amended) A method of measuring a response time associated with a request transmitted from a web browser to a web server, comprising:
deploying a script entity on said web browser, said script entity registering with said a web server to receive one or more call backs therefrom at selected points during a transaction initiated by said web server in response to the request from the web browser,
utilizing said script entity to start a clock upon transmission of said request from said web browser to said web server, and
utilizing said script entity to stop the clock, to thereby determine said response time,
upon receipt of a call back from said web server indicating completion of said transaction.